**REPORT**

Entitled

**“TEMPERATURE DISTRIBUTION IN 1-D FIN**

**SUBJECT TO VOLUMETRIC HEATING”**

*Submitted to the Department of Mechanical Engineering,*

*In Partial Fulfilment of the Requirement for the Degree of*

**Bachelor of Technology**

(Department of Mechanical Engineering)

Presented & Submitted By:

**Zeel Patel Lalithya Varshith**

**(U21ME186) (U21ME225)**

**COMPUTATIONAL FLUID DYNAMICS**

**B. TECH. 3rd Year – 5th Semester.**



(Year: 2023-24)

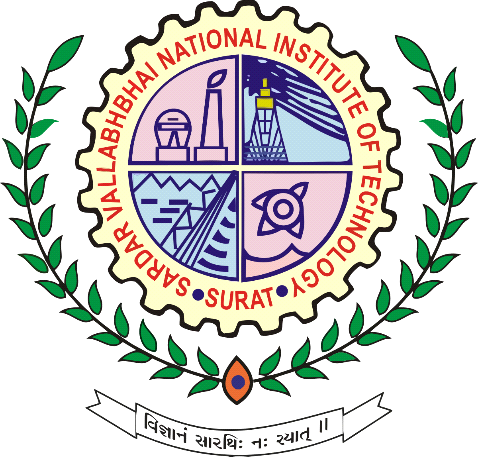
DEPARTMENT OF MECHANICAL ENGINEERING

Sardar Vallabhbhai National Institute of Technology

Surat-3950007, Gujarat, INDIA.

**SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY**

SURAT-395007, GUJARAT, INDIA



**CERTIFICATE**

This is to certify that the report entitled “Temperature distribution in 1-D fin subjected to volumetric heating” is submitted by  **ZEEL PATEL[U21ME186]**, **Varshith[U21ME225]** in partial fulfilment for the award of the degree in “Bachelor of Technology” in Mechanical Engineering during the academic year 2023-2024 of the Sardar Vallabhbhai National Institute of Technology, Surat.

SIGN-

**INDEX**

**CONTENT Pg.no**

1. INTRODUCTION/OBJECTIVE OF THE ASSIGNMENT 4
2. MODEL EQUATION 5
3. DISCRITIZATION 5
4. HAND-CALCULATION 6
5. CODE FOR TEMPERATURE VALUES 13
6. CODE FOR HEAT-PLO (GNUPLOT) 19
7. SCREENSHOT OF CODE AND CONSOLE 20
8. TEMPERATURE DISTRIBUTION SUBJECTED TO CONSTANT 23

S(VOLUMETRIC HEATING)

1. TEMPERATURE DISTRIBUTION SUBJECTED TO 27

S(VOLUMETRIC HEATING) AS AFUNCTION OF TEMPERATURE

1. TEMPERATURE DISTRIBUTION SUBJECTED TO 28

S(VOLUMETRIC HEATING) AS A FUNCTION OF DISTANCE.

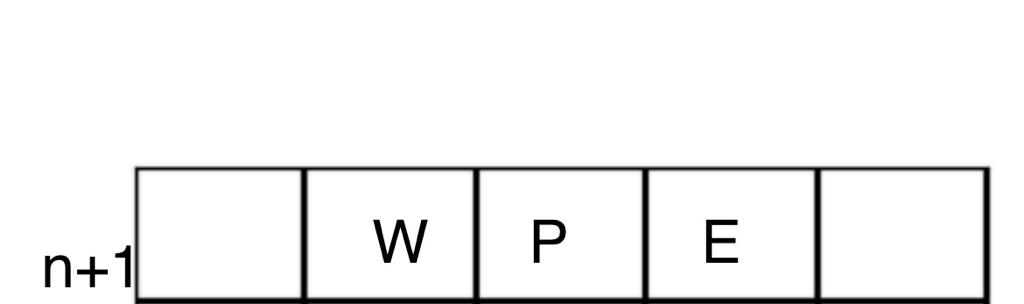
1. CONCLUSION 29

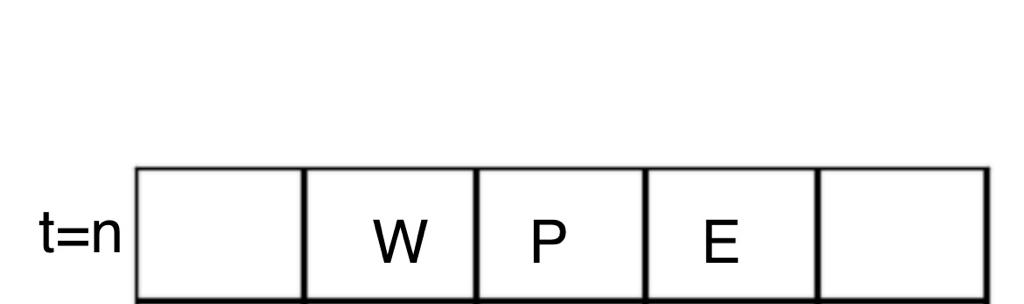
**OBJECTIVE OF THE ASSIGNMENT**

* *Develop a numerical code for obtaining unsteady temperature variation in a 1-D fin with bulk or volumetric heating. Generate results for all boundary conditions at both ends and different values of volumetric heating. The circumference of the fin is insulated. Use TDMA.*
* *The volumetric heating rate is –*

1. *Constant throughout the fin*
2. *Function of temperature*
3. *Function of x(distance from fin end)*

* *To compare the values of temperature distribuition when subjected to different values of volumetric heating rate.*





MODEL EQUATION AND BOUNDARY CONDITION

Thermal Energy conservation;

[𝑞 ̇(𝑥) − 𝑞 ̇(𝑥 + ∆𝑥)]𝐴 + S(A∆𝑥) =

Where, A = Cross-sectional area of Fin p = Perimeter of fin

S= Volumetric heating rate = Conductive Heat flux (mCT) = Change in Internal Energy

𝑞 ̇(𝑥 + ∆𝑥) = 𝑞 ̇(𝑥) − ∆𝑥( ) 𝑚𝐶𝑇 = (𝜌A∆𝑥)𝐶𝑇

𝜕(𝜌A∆𝑥𝐶𝑇) = − ∆𝑥𝐴 ( ) +S(∆𝑥𝐴)

Now, 𝑞̇(𝑥) = −𝑘

-*MODEL EQUATION*

*BOUNDARY CONDITIONS-*

Dirichlet boundary condition at x=0 = 1

x=L = 0

*INITIAL CONDITION-*

***DISCRETIZATION-***

Applying Implicit Time Marching,

General Equation-

+ S

Where,

(where )

= -

= -

= 1

 For cells(2 ≤ *i* ≤ (M-1))

+ S

For *i*=1,

+ S

Applying boundary condition at left side,

= 1

=

+ S

+ S

For *i*=M,

+ S

Applying boundary condition at right end,

= 1

+ S

(

***SAMPLE CALCULATION ( DIRICHLET BOUNDARY CONDITION)***

L=1 M=5 =1

*INITIAL CONDITION*

S= Volumetric Heating Rate= 0.01

= 1.5

= - = -0.25

= - = -0.25

= 1

5 equations for 5 unknown values-

+ S

+ S

+ S

+ S

(

(n=0)

(

Now, TDMA AX=B

=

=

Applying Row operations (Forward elimination)

1i b= lower diagonal term

d= Main diagonal term

=

Now Backward Substitution, we get,

X=

(n=1)

(

Now, TDMA AX=B

=

=

Applying Row operations (Forward elimination)

1i b= lower diagonal term

d= Main diagonal t

Now Backward Substitution, we get,

X=

(n=2)

(

Now, TDMA AX=B

=

=

Applying Row operations (Forward elimination)

1i b= lower diagonal term

d= Main diagonal term

  Now Backward Substitution, we get,

X=

We continue the iterations till RMS

***SAMPLE CALCULATION ( ROBIN’S BOUNDARY CONDITION)***

*Robin’s Boundary Condition on right side*

We get,

For *i*=M,

+ S

Applying boundary condition,

A =

B=

+ S

+ S -

 L=1 M=5 =1

h/2k =1

S= Volumetric Heating Rate= 0.01

= 1.5

= - = -0.25

= - = -0.25

= 1

5 equations for 5 unknown values-

+ S

+ S

+ S

+ S

+ S –

(n=0)

Now, TDMA AX=B

=

=

Applying Row operations (Forward elimination)

1i b= lower diagonal term

d= Main diagonal term

=

Now Backward elimination, we get,

X=

We continue the iterations till RMS

***S varies with Temperature(S’=S(1+BT))***

Equation-

+ S(1+B)

+ S

Then, we will proceed similarly as above with modified equation

WHERE B=constant

***S varies with Fin Distance(S’=S(1+BX))***

Equation-

+ S(1+BX)

WHERE X= fin distance from left side

Then, we will proceed similarly as above with modified equation

WHERE B=constant

**CODE**

//#include<iostream>

#include<stdio.h>

#include<math.h>

#include<conio.h>

#include<io.h>

main()

{

int nx;

printf("enter number of grids- ");

scanf("%d",&nx);

double x[nx+10],T[nx+10],Told[nx+10],d[nx+10],a[nx+10],b[nx+10],R[nx+10];

mkdir("SC");

int i,step,M,boundary\_choicel,boundary\_choicer;

double L,dx,A,dt,TH,TC,ap,aw,ae,a0,rms,residue,S,h,k,Ts,q,c,B;

dt = 0.01; //time-step

residue = 0.000001; //residue

B=0.002;

printf("enter the value of volumetric heating rate S= ");

scanf("%lf", &S);

printf("\nEnter 1 for constant S\nEnter 2 for S as function of temperatue\nEnter 3 for S as function of fin distance\n");

printf("Volumetric heating rate function -");

scanf("%lf",&c);

printf("\nenter the value of L= ");

scanf("%lf", &L);

printf("enter the value of TH(left)= ");

scanf("%lf", &TH);

printf("enter the value of TC(right)= ");

scanf("%lf", &TC);

printf("enter the value of A(diffusivity)= ");

scanf("%lf", &A);

printf("enter the value of Ts= ");

scanf("%lf", &Ts);

printf("\npress 1 for Dirichlet boundary condition\npress 2 for Robin's boundary condition\npress 3 for Newmann Homogeneous boundary condition\npress 4 for Newmann Non-Homogeneous boundary condition");

printf("\nboundary choice for left side - ");

scanf("%d",&boundary\_choicel);

printf("boundary choice for right side - ");

scanf("%d",&boundary\_choicer);

if (boundary\_choicel==2 or boundary\_choicer==2)

{

printf("\nenter the value of h= ");

scanf("%lf", &h);

printf("enter the value of k= ");

scanf("%lf", &k);

}

if (boundary\_choicel==4 or boundary\_choicer==4)

{

printf("enter the value of k= ");

scanf("%lf", &k);

}

if (boundary\_choicel==4 or boundary\_choicer==4)

{

printf("enter the value of q= ");

scanf("%lf", &q);

}

dx = L/nx;

x[1] = 0.5\*dx;

x[0] = 0.0;

x[nx+1] = L;

for(i=2;i<=nx;i++)

x[i] = x[i-1] + dx;

//----------coefficients-------------//

if(c==2)

{

ap = (1+((2\*A\*dt)/(dx\*dx))-B\*S);

}

else

ap = (1+((2\*A\*dt)/(dx\*dx)));

aw = (-1)\*((A\*dt)/(dx\*dx));

ae = (-1)\*((A\*dt)/(dx\*dx));

a0 = 1.0;

//-------initialisation----------//

for(i=1;i<=nx;i++)

{

T[i] = 0.0;

Told[i]=0.0;

}

//---------iterative calculation with TDMA-------------//

step = 1;

do

{ for(i=1;i<=nx-1;i++) //above diagonal

a[i] = ae;

for(i=2;i<=nx;i++) //below diagonal

b[i] = aw;

if (boundary\_choicel==1)

d[1] = (ap-aw); //main diagonal

else if (boundary\_choicel==2)

d[1]= (ap+((2\*k+h\*dx)/(2\*k-h\*dx)\*aw));

else if (boundary\_choicel==3)

d[1]=(ap+aw);

else if (boundary\_choicel==4)

d[1]=(ap+aw);

if (boundary\_choicer==1)

d[nx] = (ap-ae);

else if (boundary\_choicer==2)

d[nx] = (ap+((2\*k-h\*dx)/(2\*k+h\*dx)\*ae));

else if (boundary\_choicer==3)

d[nx]=(ap+ae);

else if (boundary\_choicer==4)

d[nx]=(ap+ae);

for(i=2;i<=nx-1;i++)

d[i] = ap;

if(c=1)

{

if (boundary\_choicel==1)

R[1] = ((a0\*T[1])-(2\*TH\*aw)+S); //B matirx in AX=B //main diagonal

else if (boundary\_choicel==2)

R[1] = ((a0\*T[1])+((2\*h\*dx)/(2\*k-h\*dx)\*aw\*Ts)+S);

else if (boundary\_choicel==4)

R[1] = ((a0\*T[1])-(aw\*q\*dx/(k))+S);

else if (boundary\_choicel==3)

R[1] = ((a0\*T[1])+S);

if (boundary\_choicer==1)

R[nx] = ((a0\*T[nx])-(2\*TC\*ae)+S);

else if (boundary\_choicer==2)

R[nx] = ((a0\*T[nx])-((2\*h\*dx)/(2\*k+h\*dx)\*ae\*Ts)+S);

else if (boundary\_choicer==4)

R[nx] = ((a0\*T[nx])+(ae\*q\*dx/(k))+S);

else if (boundary\_choicer==4)

R[1] = ((a0\*T[nx])+S);

for(i=2;i<=nx-1;i++)

R[i] = a0\*T[i]+S;

}

else if (c==3)

{

if (boundary\_choicel==1)

R[1] = ((a0\*T[1])-(2\*TH\*aw)+S\*B\*x[1]); //B matirx in AX=B //main diagonal

else if (boundary\_choicel==2)

R[1] = ((a0\*T[1])+((2\*h\*dx)/(2\*k-h\*dx)\*aw\*Ts)+S\*B\*x[1]);

else if (boundary\_choicel==4)

R[1] = ((a0\*T[1])-(aw\*q\*dx/(k))+S\*B\*x[1]);

else if (boundary\_choicel==3)

R[1] = ((a0\*T[1])+S\*B\*x[1]);

if (boundary\_choicer==1)

R[nx] = ((a0\*T[nx])-(2\*TC\*ae)+S\*B\*x[nx]);

else if (boundary\_choicer==2)

R[nx] = ((a0\*T[nx])-((2\*h\*dx)/(2\*k+h\*dx)\*ae\*Ts)+S\*B\*x[nx]);

else if (boundary\_choicer==4)

R[nx] = ((a0\*T[nx])+(ae\*q\*dx/(k))+S\*B\*x[nx]);

else if (boundary\_choicer==4)

R[1] = ((a0\*T[nx])+S\*B\*x[nx]);

for(i=2;i<=nx-1;i++)

R[i] = a0\*T[i]+S\*B\*x[i];

}

for(i=2;i<=nx;i++) //forward sweep

{

d[i] = (d[i] - ((b[i]/d[i-1])\*a[i-1]));

R[i] = (R[i] - ((b[i]/d[i-1])\*R[i-1]));

}

T[nx] = R[nx]/d[nx]; //backword sweep

for(i=nx-1;i>0;i--)

T[i] = ((R[i] - (a[i]\*T[i+1]))/d[i]);

T[0] = TH;

T[nx+1] = TC;

double sum = 0.0; //rms calulation

for(i=1;i<=nx;i++)

{

sum = sum + pow((Told[i]-T[i]),2);

Told[i]=T[i];

printf("%d\t%lf\t%lf\n",step,rms,T[i]);

}

rms = sqrt(sum/nx);

if (step%1==0)

{

char filename[1000]; //////size of charecter

FILE\*f2;

sprintf(filename,"SC\\%d.dat",step);

f2=fopen(filename,"w");

fprintf(f2,"VARIABLES=X,T\n");

fprintf(f2,"\nZONE T=\"0\" i=%d, ZONETYPE=ORDERED,DATAPACKING=POINT\n",nx);

for (i=1;i<=nx;i++)

{{fprintf(f2,"%f\t%.10f\n ",x[i],T[i]);}}

fclose(f2);

}

step++;

}

while(rms>residue);

//printf("%d",step);

//-------------file writing------------//

FILE \*f1;

//for(i=0;i<=nx;i++)

{f1 = fopen("temp.dat","w");

fprintf(f1,"VARIABLES=""X"",""TEMPERATURE""\nZONE I=%d ZONETYPE=ORDERED DATAPACKING=POINT\n",nx+2);

for(i=0;i<=nx+1;i++)

fprintf(f1,"%lf\t%lf\n",x[i],T[i]);

}

fclose(f1);

return 0;

}

**GNUPLOT CODE**

maxIterations=400

# Specify the output file format and name

set terminal pngcairo

set output "temperature\_plots.png"

# Define plot style and labels

set xlabel "X"

set ylabel "Temperature"

set title "Temperature Variation over Iterations S(x)"

# Set a smaller legend size

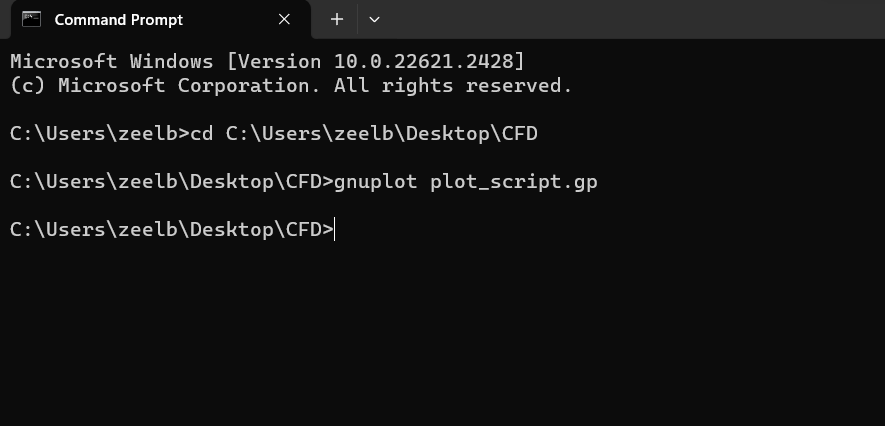
#set key samplen 1 spacing 1

# Plot data from multiple files using a loop

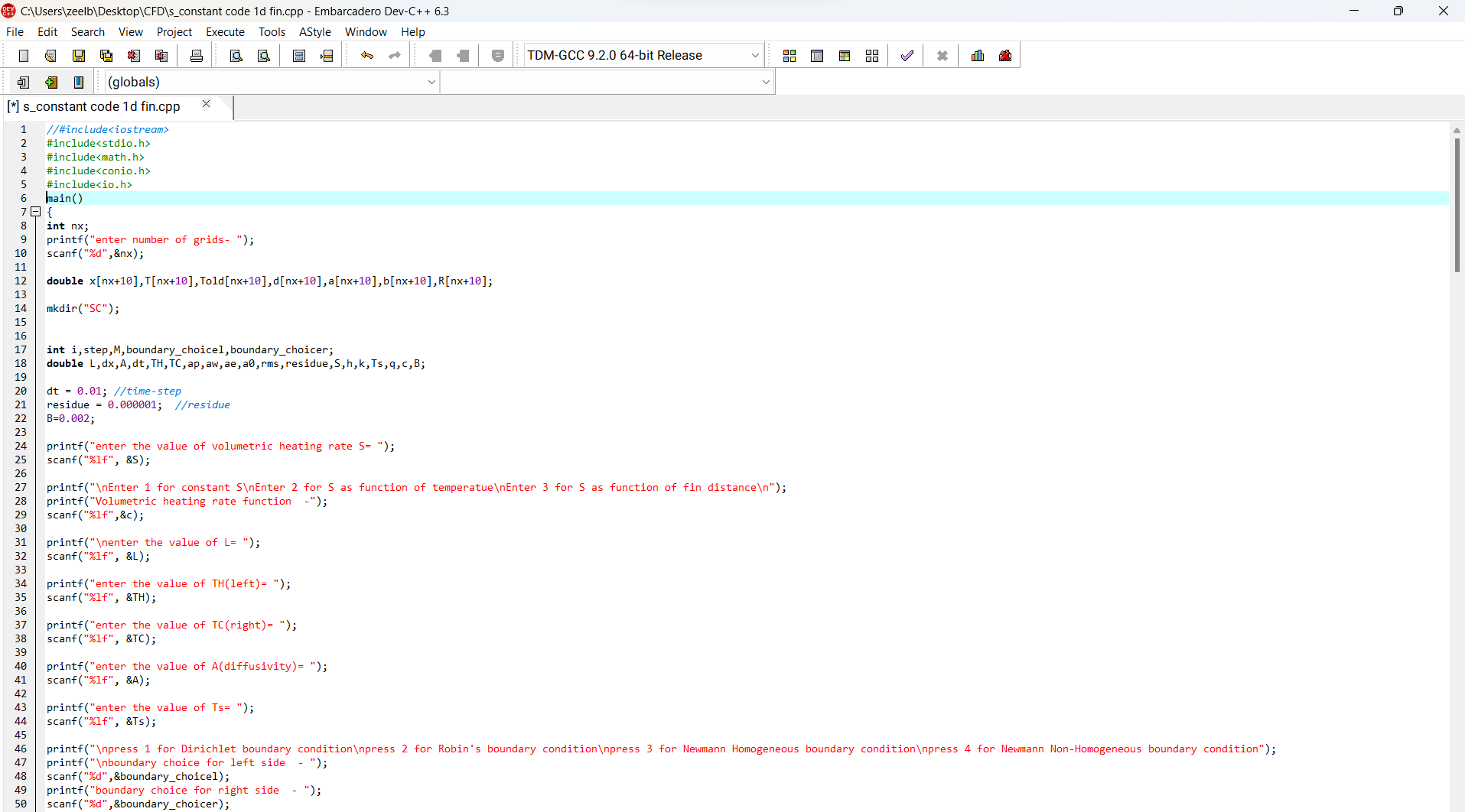
#plot for [i=0:maxIterations] 'TEST/'.i.'.dat' using 1:2 with lines title 'S=0.0'.i

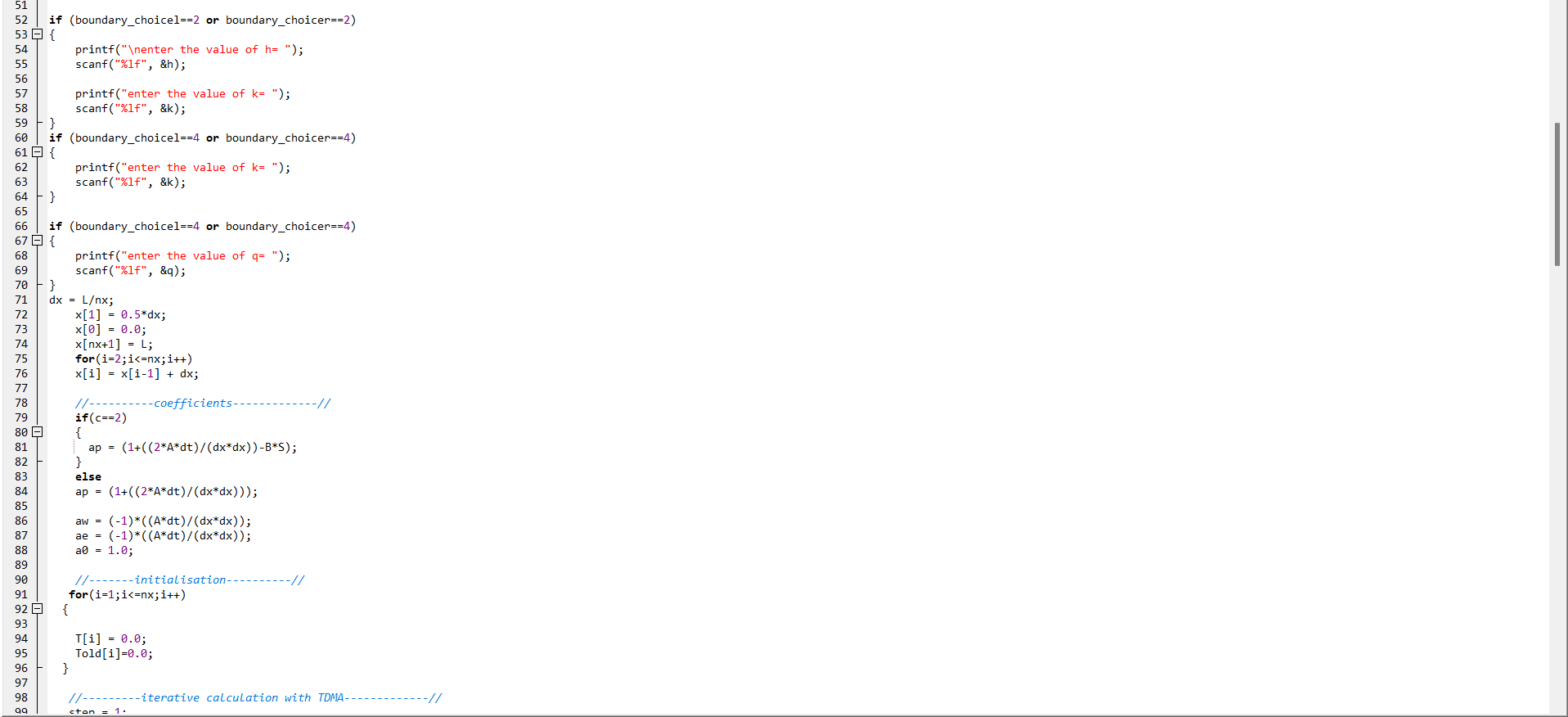
plot for [i=1:maxIterations:10] 'SX/'.i.'.dat' using 1:2 with lines notitle

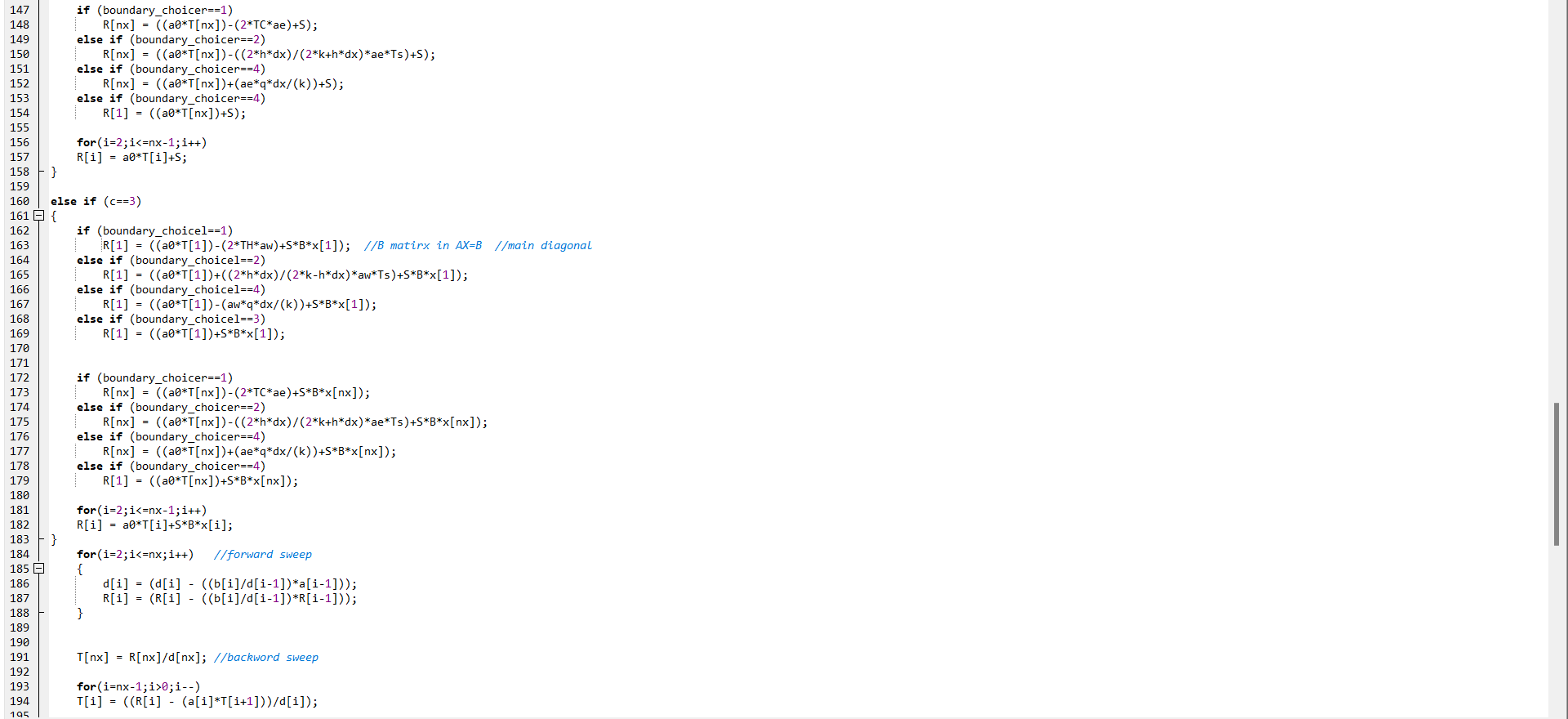
THIS CODE IS EXECUTED IN COMMAND PROMPT BY SPECIFYING THE LOCATION OF FOLDER WHERE TDMA VALUES ARE STORED,

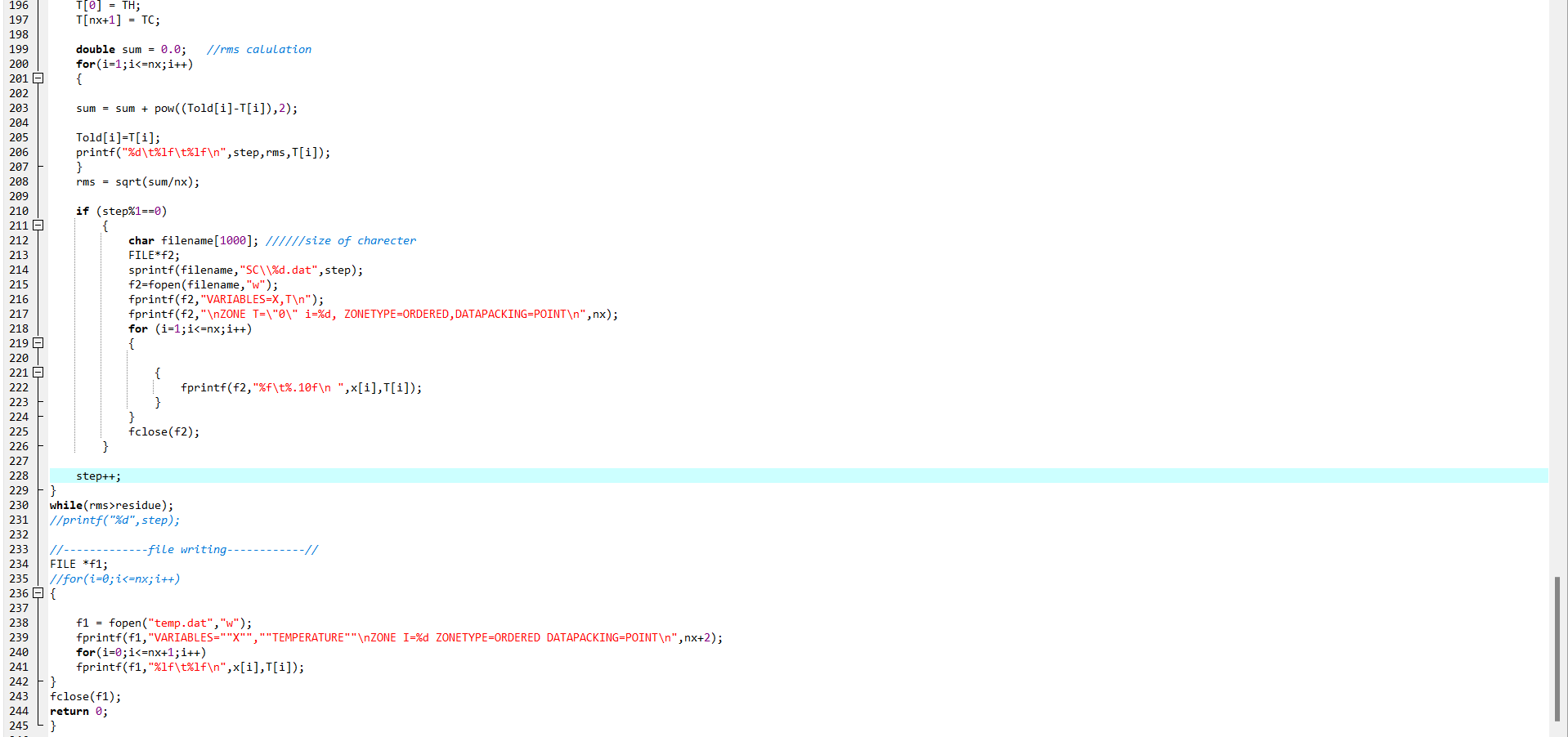


**CODE SCREENSHOT**

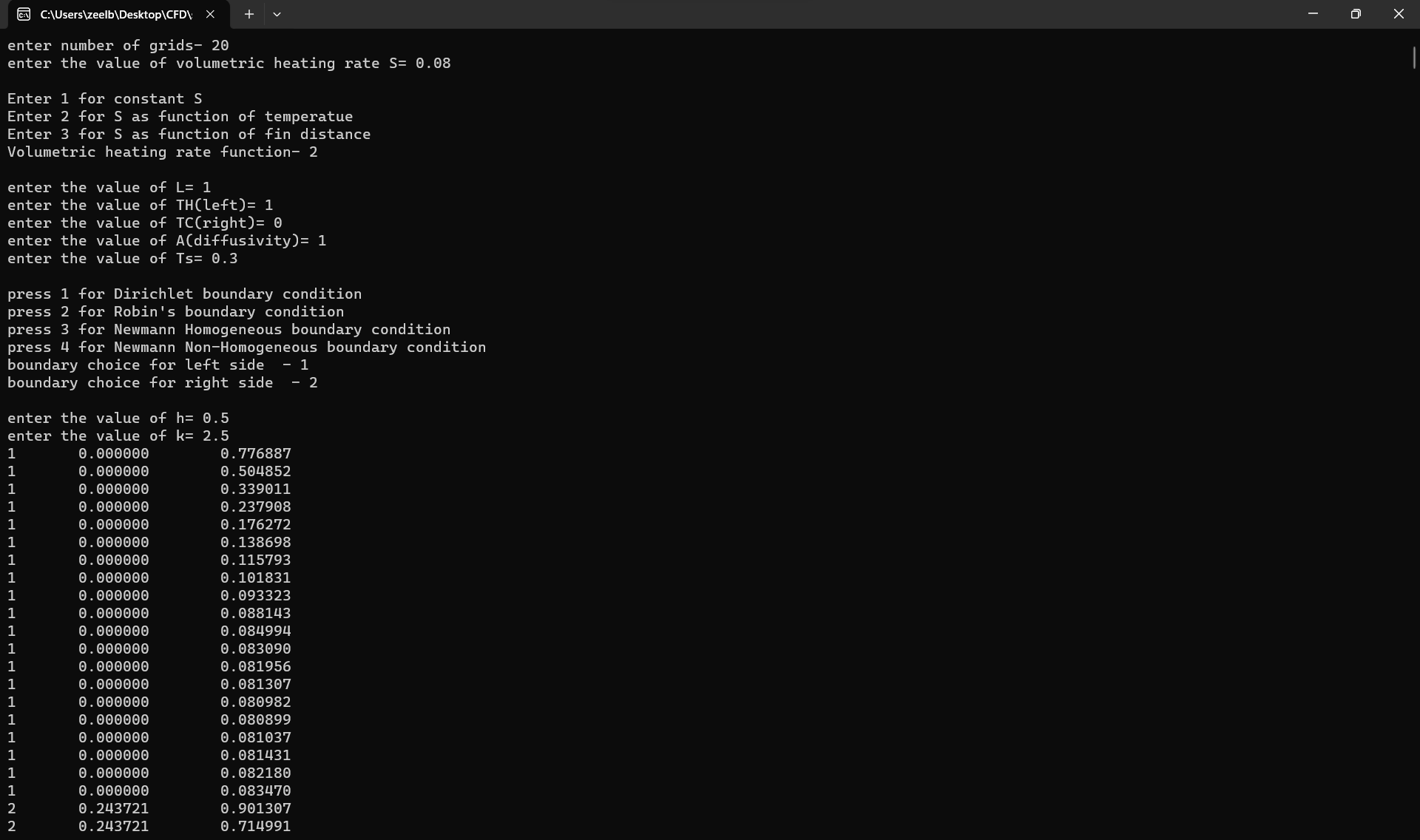
****

****

****

****

**CONSOLE SCREENSHOT**

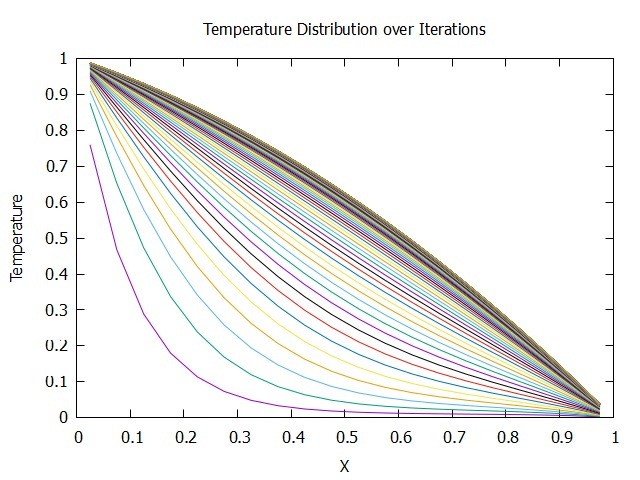
****

Here, all the parameters are entered by user like-

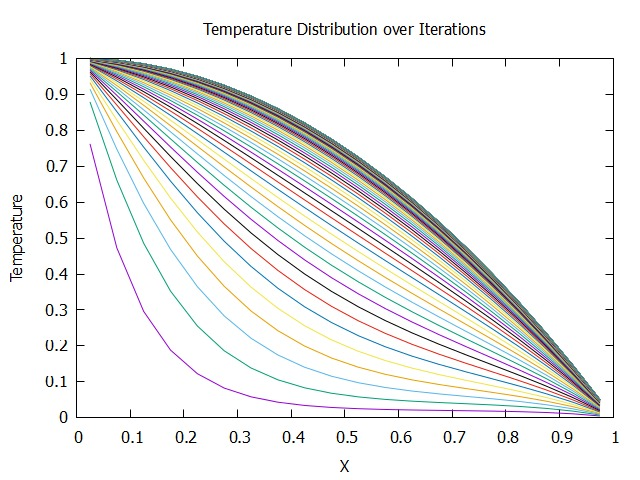
Boundary conditions, Volumetric heating rate function and other parameters as shown in the above screenshot.

***DIRICHLET BOUNDARY CONDITION***

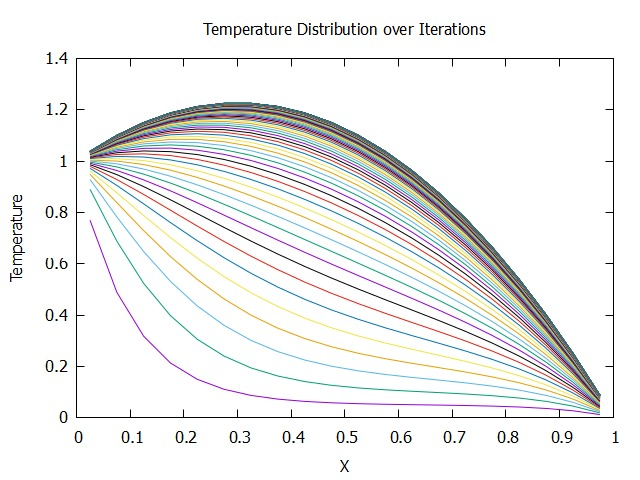
***[S=VOLUMETRIC HEATING RATE]***



S=0.01 AND 116 ITERATIONS



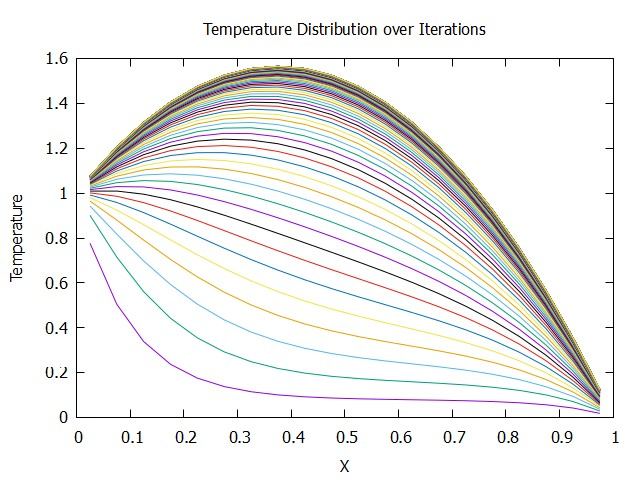
S=0.02 AND 118 ITERATIONS



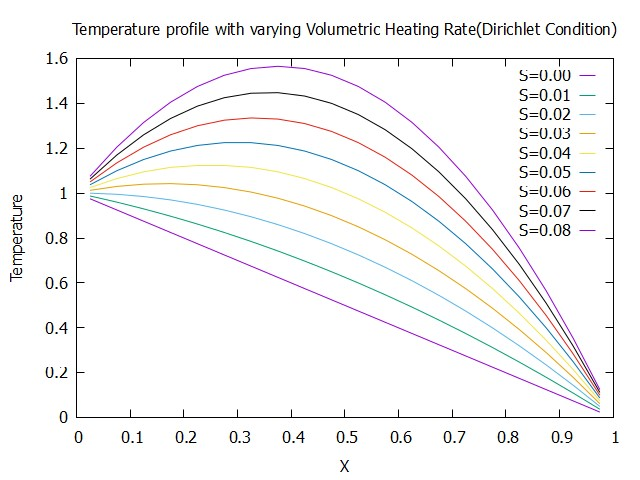
S=0.05 AND 122 ITERATIONS



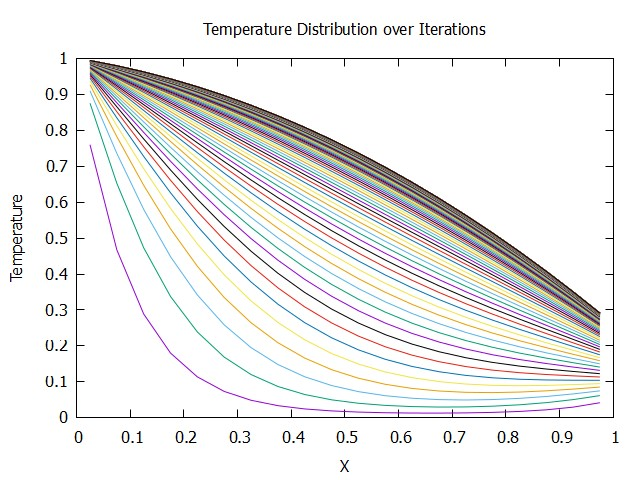
S=0.06 AND 123 ITERATIONS



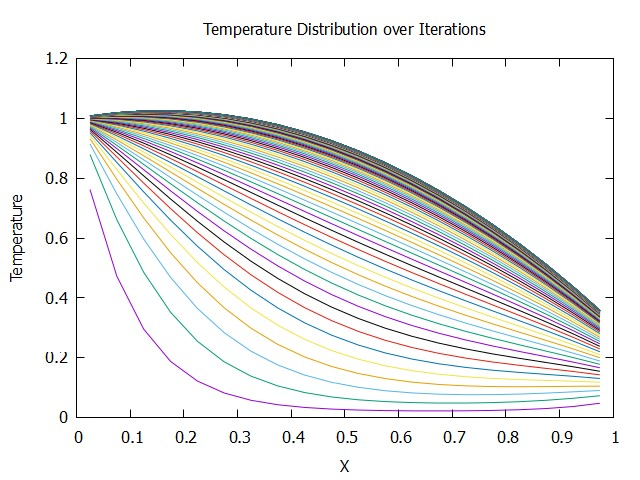
S=0.08 AND 125 ITERATIONS



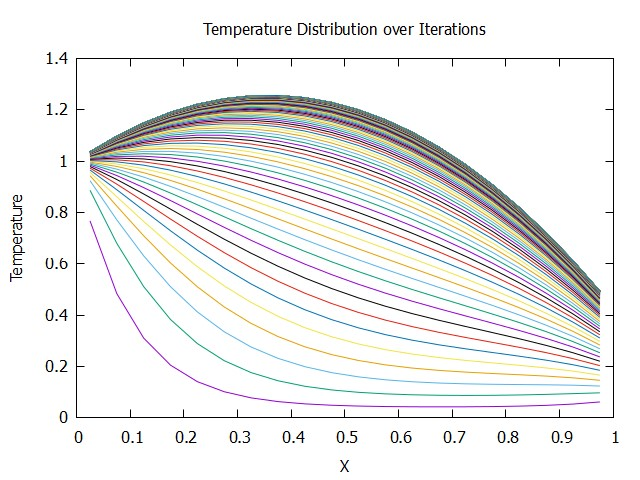
**ROBIN’S BOUNDARY CONDITION**



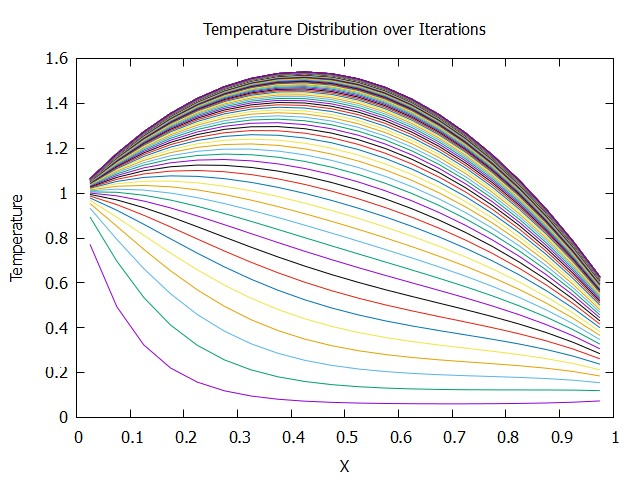
S=0.01 AND 144 ITERATIONS



S=0.02 AND 146 ITERATIONS



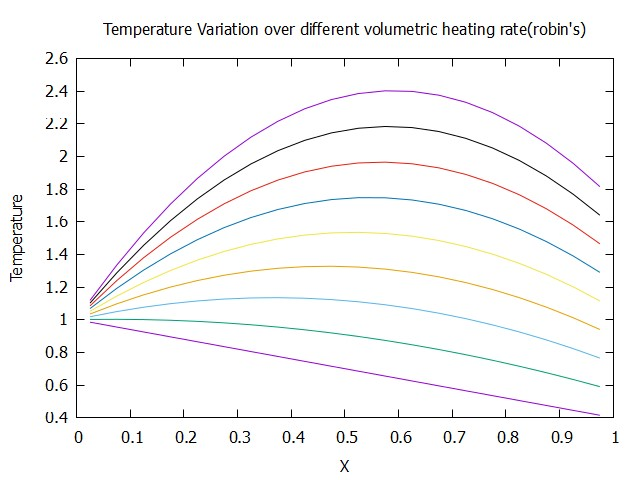
S=0.04 AND 150 ITERATIONS



S=0.06 AND 153 ITERATIONS

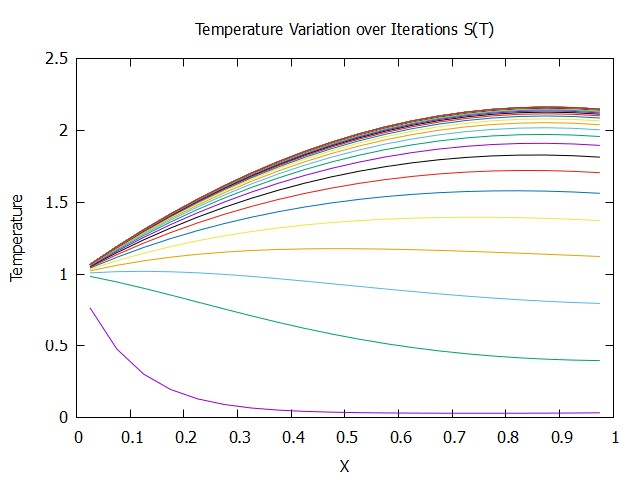
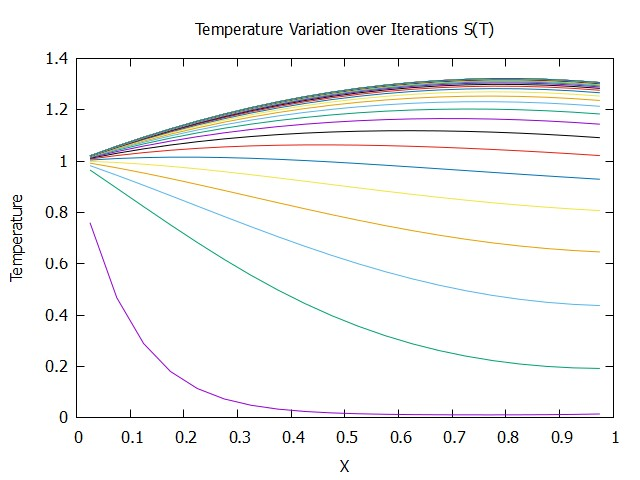


S=0.08 AND 155 ITERATIONS

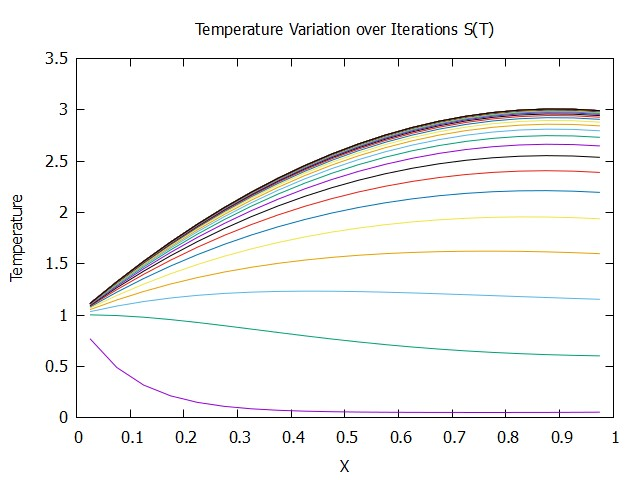
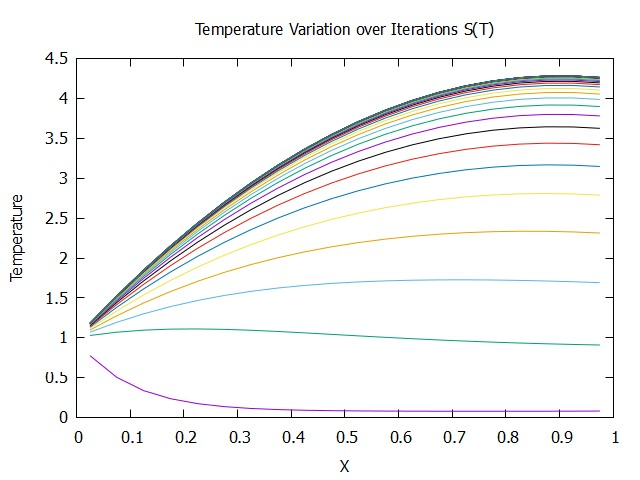


***S FUNCTION OF T( B=0.002)***

***S’=S(1+BT)***



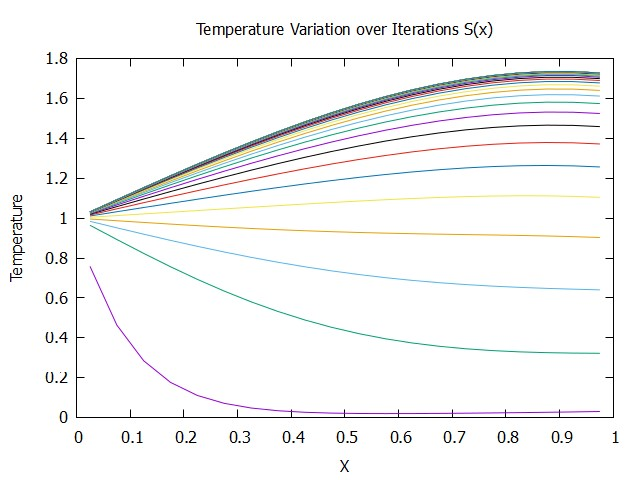
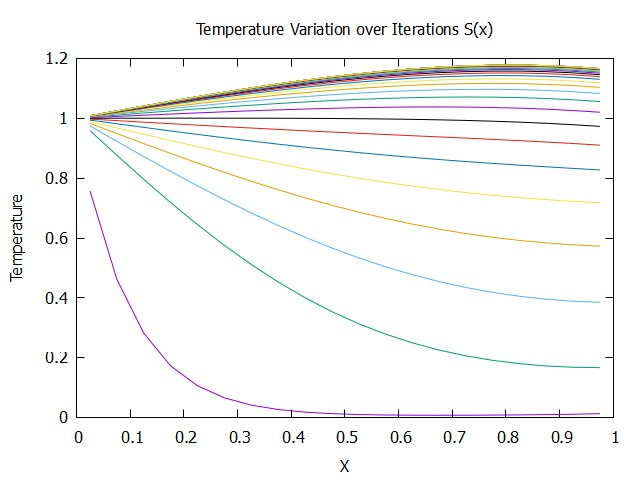
S=0.01 S=0.03

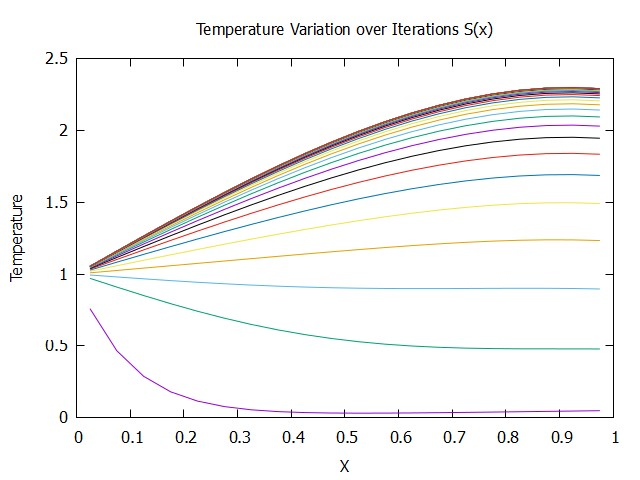
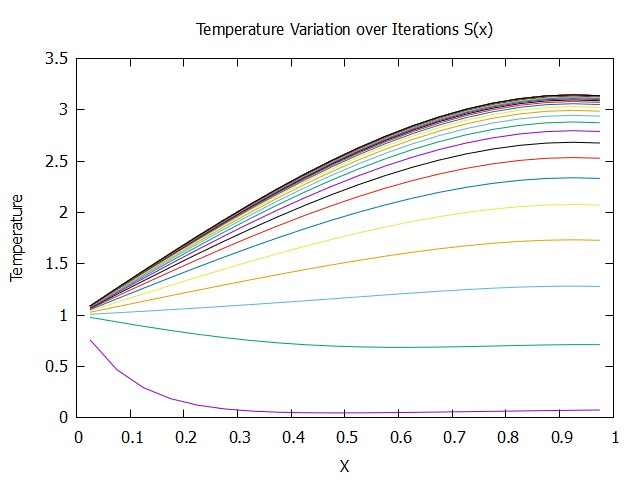
S=0.05 S=0.08

**S FUNCTION OF X(B=0.02)**

**S’=S(1+BX)**



S=0.01 S=0.03

S=0.05 S=0.08

*CONCLUSION*

* As the volumetric heating rate increases, the maxima shifts further upwards.
* Use of Dirichlet boundary condition and circumferential insulation had a profound impact on temperature distribution.
* If the right side of the 1D fin is subjected to a Robin boundary condition while maintaining the Dirichlet condition on the left. The transition from a Dirichlet to a Robin boundary condition on one side introduces a more complex thermal interaction, allowing for a heat transfer coefficient to influence the temperature profile at that boundary.
* The shift in the location of the temperature maximum with increasing volumetric heating rate would typically be more pronounced in the Robin boundary condition case compared to the Dirichlet case.
* In the Dirichlet boundary condition, the temperature at the boundary is fixed, and the heat accumulates within the fin, leading to a more gradual change in temperature distribution as the volumetric heating rate increases.
* In the Robin boundary condition, the right-side temperature is influenced by the heat transfer coefficient and the temperature difference between the fin and its surroundings. As the volumetric heating rate increases, the higher heat generation causes a more significant temperature difference, which in turn affects the right-side temperature more significantly.
* As the temperature varies with temperature and distance, we observe that peak(maxima) is flattered compared to constant heating rate because we observe the steady rise in the values of the volumetric heating rate.